

webpack(1)

1. 什么是webpack

webpack => 模块打包机 => exports / module.exports

它可以分析项目结构，找到JavaScript模块，以及其他浏览器不能运行的扩展语言

例如：（less）并将其转换和打包为合适的格式供浏览器使用

2. 为什么使用webpack

把js,css img 等（模块）打包；减少请求；

3. 安装webpack

还是通过npm下载

第一步该用什么命令

npm init => package.json

第二步 安装webpack

=====>全局安装（不推荐=>原因 =>框架更新太快）

npm i (install) -g webpack@(版本号)

=====> 局部安装（推荐）--save-dev 开发的模式去使用

npm i --save-dev webpack@（版本号）学的（3.5.6）

用终端命令 实现 webpack 打包(webpack 模块)（入口）（出口）

node_modules\.bin\webpack app\main.js public\build_main.js

用终端操作终归不好管理

那么，webpack提供了一个配置文件 名字叫做webpack.config.js(规定)

然后可以直接使用node_modules\.bin\webpack

来进行打包 => 他会搜索是否存在webpack.config.js

那么，node_modules\.bin\webpack = > 还是很长

能不能简单好记一点呢

在package.json里面的script这个对象

是一个自定义指令的位置

我们可以简化命令实现打包

npm 直接 key 是由限制的

如果不是npm自带的关键字，那么 你需要 => npm run key

webpack 的强大功能 调试更简单

source maps

打包之后的文件 如果配置了 source maps 可以看到依赖的代码

=====>到底在哪里配置呢？？？？？

webpack.config.js

devtool:'eval-source-map'

// eval-source-map 偏中小型的一种

使用eval 打包源文件模块，在同一个文件中

生成一个干净完整source map。这个文件可以在不影响构建速度的前提下生成完整的sourcemap

但对打包后输出的JS文件的执行具有性能和安全的隐患。在开发阶段这是一个非常好的选项，在

生产阶段一定不要启用这个选项。

对小到中型的项目中，eval-source-map是一个很好的选项，在此强调你只应该在开发阶段使用它

我们继续对上文新建的webpack.config.js, 进行配置

代码热更新 => 创建自己的开发测试服务器

webpack提供一个本地开发服务器，这个服务器是基于node.js

不过这个模块需要下载 并且依赖一下

cnpm i --save-dev webpack-dev-server@2.9.5

webpack-dev-server => 3.0以上版本是为webpack4.0服务的

我们用2.9.5版本

命令: npm run server

webpack又一个强大功能

---loaders =====>

通过使用不同的loader，webpack 有能力调用外部的工具和模块，

实现对不同格式文件的处理，比如说把es6=>es5;转化成现代浏览器兼容的js文件

=====>因为babel 一直就在于这个兼容性转义的，所以用babel

//https://www.babeljs.cn/

====> 下载babel的处理模块包

babel的安装和配置

babel的三个核心模块

babel-loader ===》让webpack知道babel如何运行

babel-core ===》让babel知道如何解析代码

babel-preset-env ===》可以根据不同的环境转换代码

cnpm i --save-dev babel-loader babel-core babel-preset-env

这个改一下版本: cnpm i --save-dev babel-loader@7.1.1